

# 지그비 네트워크에서 효율적인 분산형 주소 할당 방법

An Efficient Distributed Address Assignment Mechanism for ZigBee

박성진 (세종대학교, 석사과정)      이은주 (ETRI)      류재홍 (ETRI)      주성순 (ETRI)      김형석 (세종대학교, 교수)

Key Words : ZigBee, Distributed addressing, Borrowing

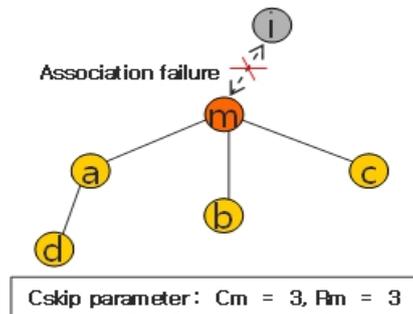
## 목 차

- I. 서론
- II. 분산형 차용식 주소 할당방식
  - 1. 주소 할당
  - 2. 주소 차용
- III. 성능 분석
- IV. 결론
- 3. 테이블 관리

### I. 서론

IEEE 802.15.4[1]에서 네트워크 및 보안계층, 네트워크 토폴로지를 확장한 지그비 기술은 라우팅 알고리즘과 Adhoc, Star, Mesh 네트워크 등 다양한 네트워크 구성을 지원한다. 현재 지그비 프로[2]에서는 네트워크 어드레스 할당 방식으로 두 가지를 제시하고 있으며, 이는 분산 주소 할당 방식과 확률적인 주소 할당 방식이다. 이 두 가지 어드레스 할당의 공통적인 특징은 특정 중앙 노드의 도움 없이 연결되는 부모 노드 스스로 어드레스를 결정하여 할당한다는 점이다. 분산 주소 할당 방식은 계층 구조의 트리를 기반으로 하고 있으며 확률적인 주소 할당 방식은 무작위 주소 할당 후 중복 검사를 수행한다. 이외에도 MANET이나 메쉬네트워크 등에서 일반적으로 제안되는 주소 할당 방식을 크게 분류하면 계층적 어드레스 할당, 중앙 집중형 어드레스 할당[3][4], 클러스터별 어드레스 할당, 무작위 주소 할당[5], 선형 주소 할당[6], 그리고 이들의 혼합형[7] 등으로 분류할 수 있다.

트리 기반의 지그비 네트워크에서는 각 노드들을 다른 노드들과 구별될 수 있도록 하는 유일성이 보장된 주소할당 방식을 필요로 한다. 이러한 주소할당 방식을 통해 네트워크 전 노드들에게 형성된 주소체계는 라우팅에 용이하여야 한다. 현재, 지그비 스펙의 Cskip을 이용한 어드레싱 방식은 미리 최대 트리의 깊이(Lm)와 최대 자식 수(Cm)와 최대 라우터수(Rm)를 지정해놓고 시작된다. 이는 새로운 노드가 네트워크에 진입할 때, 어드레스의 유일성을 보장하면서 별도의 라우팅 테이블이나 경로 검색 과정에 대한 요구 없이 단지 주소 정보만을 이용하여 라우팅을 간소화한다는 이점을 가진다. 하지만, 파라미터의 설정이 노드들의 지형적 분산과 맞지 않으면, 사용하지 않는 주소 공간이 많이 남아 있다 할지라도 주소를 할당 받지 못하는 노드들이 생겨나는 문제점을 가진다. 더욱이 이러한 문제를 해결하기 위한 자동적인 파



<그림 1> 분산 주소 할당 방식의 문제점

라미터 설정을 산출해내는 방법 등은 현재까지 없기 때문에 Cskip을 이용한 어드레싱 방식의 약점으로 남아있다. 이에 본 논문에서는 노드들이 지형적 분산과 맞지 않더라도 낭비되는 여분의 주소를 최소화하고 노드에 유일성을 보장할 수 있는 주소할당 방식을 제안하고 더불어 이를 이용한 라우팅 방식을 제안한다.

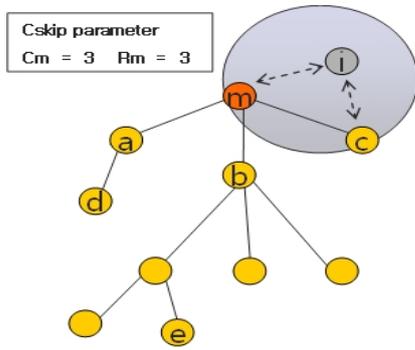
본 논문의 구성은 다음과 같다. 2장에서는 제안하는 주소 할당 방법의 기본적 동작에 대해서 기술하고, 3장에서는 제안된 방법을 기존의 방법들과 비교해 그 성능을 분석한다. 마지막으로 4장에서 본 연구의 결론에 대해서 기술한다.

### II. 분산형 차용식 주소 할당방식

#### 1. 주소 할당

신규 노드가 지그비 네트워크에 접근 할 때, 노드는 주소를 할당받기 위해 스캐닝을 통해 부모 노드를 찾아야 한다. 그림 2에서 보이는 것처럼 만일, 신규 노드가 중첩된 위치에 놓여 있다면, 노드는 주변 노드들 중, 초기에 Cskip 파라미터로 정

의된  $C_m$ 의 여유가 있는 부모 노드로 선택하도록 한다. 이를 위해 각 주변 노드들은 자신의  $C_m$  여유분을 알리기 위해 비콘 프레임 안에 자식 노드의 개수를 포함하여 방송한다.



<그림 2> 신규 노드 진입

주소 요청을 받은 부모 노드는 자식 노드의 여유가 있을 경우 즉시 주소를 할당하며, 주소의 여유가 없는 경우 이웃 노드들에게 주소를 빌리기 위한 메시지를 브로드캐스트한다. 본문에서는 이러한 브로드캐스트를 받은 노드들을 대응노드라 칭한다. 이 메시지를 받은 노드들은 자신이 빌려 줄 수 있는 주소와 자식의 수를 응답 메시지에 포함하여 요청 노드에 전송하며, 여분의 주소가 없는 경우에는 응답 메시지를 전송하지 않는다. 일정 시간을 기다린 후 응답 메시지를 받은 부모 노드는 우선순위를 고려하여 차용할 주소를 선택하며 이때 우선순위는 다음과 같다.

- ① 부모 노드 또는 루트 노드로 도달하는 경로 상에 위치한 노드로부터 전송된 응답 메시지를 우선 수용한다.
- ② 부모 노드에 여유 주소가 없는 경우 가장 적은 자식 수를 가지고 있는 노드를 선택한다.
- ③ 2번 우선순위가 동일하다면 주소가 가장 큰 것을 선택한다.

이 후 선택한 차용 주소의 부모 노드에게 주소가 선택되었음을 메시지를 통해 알린다. 이 메시지를 받은 노드는 자신의 라우팅 테이블을 갱신한다.

## 2. 주소 차용

주소 차용 요청 메시지를 받은 대응 노드는 식 (1)의 계산방식을 통해 알맞은 차용 주소를 구한 후, 응답 메시지에 포함하여 요청 노드로 전송한다.

주소를 빌려 줄 것을 요청 받은 대응 노드는 할당 가능한 마지막 주소를 계산한다. 마지막 주소란, 자식 노드에게 할당하거나 타 노드에게 빌려 주지 않은 주소 중에서 가장 큰 주소 값을 의미한다. 차용 주소 계산은 식 (1)과 같다.

$$B_{FFD\_Addr} = My\_Addr + (Rm - B) \times Cskip(d) + 1$$

$$B < Rm \dots (1)$$

여기서,  $Cskip(d)$ 은 기존 지그비 스펙의 분산 주소 할당 방식에 사용되는 알고리즘이며,  $B$ 는 대여중인 주소의 개수를 의미한다. 이 둘을 이용하여, 최종적으로 할당 가능한 차용 주소인  $B_{FFD\_Addr}$ 를 산출해 낼 수 있다. 신규 노드는 앞으로

할당 가능한 Address Block {  $B\_Addr + 1$  to  $B\_Addr + Cskip(d) - 1$  }을 확보하며, 이 영역을 기존의 분산 주소 할당 방식을 이용하여 할당한다. 한편, 대응노드는 일정 시간이 지난 후 자신이 빌려준 주소가 선택되었음을 통보 받으면, 자신의 Address Block내에 가장 큰 주소를 나타내는 변수 ( $Addr\_Block$ )를 갱신한다.  $Addr\_Block$ 의 갱신은  $B_{FFD\_Addr}$ 의 Address block의 크기만큼을 차감하는 것이며  $Addr\_Block - Cskip(d)$ 의 계산을 통해 이뤄진다.

## 3. 테이블 관리

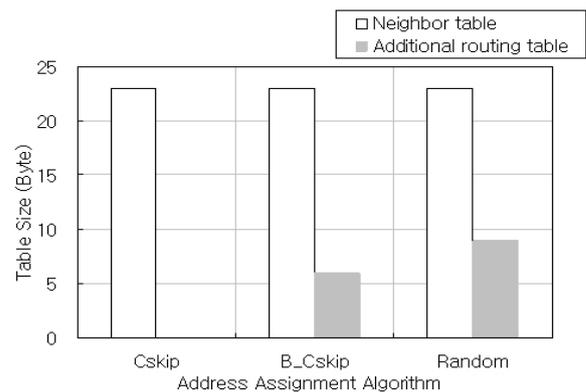
차용 주소가 사용 될 거라는 확인 메시지를 전달 받은 대응 노드는 라우팅 테이블에 요청 노드의 주소와 차용주소를 기록한다. 아래 표는 라우팅 테이블 예를 보여준다.

<표 1> 라우팅 테이블

	주소	빌린주소 여부	빌려준주소 여부
1번자식노드	1번자식노드 주소	No	No
2번자식노드	2번자식노드 주소	No	Yes (빌려줬음)
3번자식노드	3번자식노드 주소	Yes(빌린것)	No

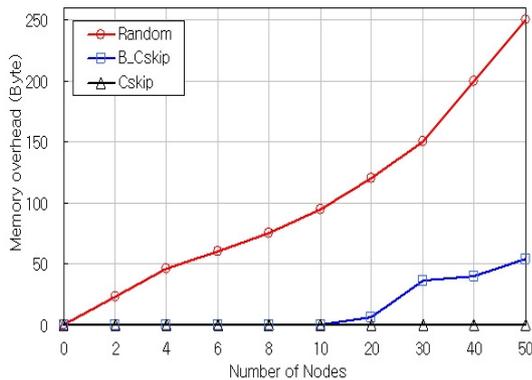
## III. 성능 분석

본 장에서는 제안하는 주소 할당 방식의 성능을 확인하기 위해 시뮬레이션을 이용하여 모의실험을 수행하였다. 제안된 방법( $B\_Cskip$ )은 분산 주소 할당 방식( $Cskip$ )과 무작위 주소 할당 방식(Random)과 비교하였다.



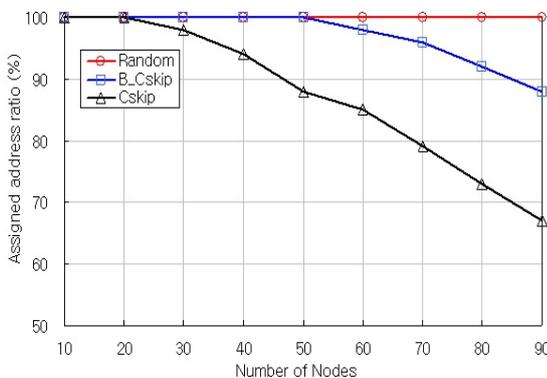
<그림 4> 알고리즘별 테이블 크기

그림 4는 주소 할당 방식에 따른 노드가 가지는 최초 테이블의 크기를 보여준다. 노드는 최초 테이블을 생성할 때 주소 할당 방식에 관계없이 지그비 스펙에서 제시하는 이웃 테이블을 가지는 것을 알 수 있다. 이와 별도로 추가 테이블은 제안하는 주소 할당 방식과 무작위 주소 할당 방식을 사용하였을 때 요구되며, 제안된 주소 할당 방식이 무작위 주소 할당 방식에 비해 작은 추가 테이블을 가지는 것을 알 수 있다.



<그림 5> 노드수 증가에 따른 라우팅 테이블 오버헤드

그림 5는 네트워크 전체의 노드수가 증가함에 따라 개별 노드의 라우팅 테이블에 발생하는 메모리 오버헤드를 나타낸다. 그림 4의 결과에서, 최초 라우팅 테이블의 크기는 무작위 주소 할당 방식이 제안된 주소 할당 방식과 큰 차이를 보이지 않는다. 반면에, 위 그래프에서는 무작위 주소 할당 방식을 사용하는 경우 노드수가 증가함에 따라 개별 노드의 라우팅을 위한 메모리 오버헤드가 급격히 증가하는 것을 볼 수 있다. 이에 비해 제안된 주소 할당 방식은 적은 메모리 사용을 나타내는데, 이는 주소 차용이 발생한 경우에만 테이블을 업데이트하기 때문이다.



<그림 6> 노드수 증가에 따른 주소 할당률

그림 6은 노드수 증가에 따른 성공적인 주소 할당 비율을 나타낸다. 분산 주소 할당 방식은 불규칙적으로 배치되거나 또는 추가하는 노드의 수가 증가 할수록 주소 할당에 성공하는 비율이 현저하게 낮아지는 것을 알 수 있다. 이에 비해 제안된 주소 할당 방식은 높은 주소 할당률을 보이며, 이는 제안한 주소 할당 방식이 노드의 지형적 배치에 있어 분산 주소 할당 방식에 비해 영향을 덜 받는다는 것을 나타낸다. 뿐만 아니라, 주소 할당의 성공률이 높다는 것은, 낭비되는 주소가 적다는 것을 의미하며, 이는 제안된 주소 할당 방식이 분산 주소 방식에 비해 효율적인 주소 사용임을 보인다.

#### IV. 결론

지그비 네트워크에서 분산 주소 할당 방식의 문제점을 보완하기 위해 본 논문에서는 기존 방식에 주변 노드의 주소를 차용하여 할당하는 방법을 제안하였다. 차용 주소의 사용은 노드의 지형적 배치에 있어서, 분산 주소 할당 방식보다 유연하게 대처 할 수 있으며, 낭비되는 여분의 주소 또한 줄일 수 있다. 뿐만 아니라, 주소 충돌 검사를 수행하는 무작위 주소 설정 방식에 비해 상대적으로 빠른 주소 설정 시간과 적은 통신 오버헤드를 가진다. 제안된 방법은 네트워크 계층에서 주소 할당에 있어 가장 큰 요구 사항인 주소의 유일성을 보장하며, 트리 기반의 지그비 네트워크에서 기존 방식에 비해 효율적인 주소 할당 방식임을 보인다.

#### 참고 문헌

- [1] Part 15.4 : Wireless medium access control(MAC) and physical layer(PHY) specifications for low-rate wireless personal area networks(LR-WPANs), IEEE Std. 802.15.4 ,2006
- [2] ZigBee specification. ZigBee Alliance, January 2008.
- [3] Mesut Gunes and Jorg Reibel, "An IP addresss Configuration Algorithm for Zeroconf. Mobile Multihop Ad-hoc Networks", In Proceedings of the International Workshop on Broadband Wireless Ad-Hoc Networks and Services, September 2002.
- [4] Yuan Sun and Elizabeth M. Belding-Royer, "Dynamic Address Configuration in Mobile Ad hoc Networks", UCSB Technical Report 2003-11, June 2003.
- [5] Hongbo Zhou, Lionel Ni, Matt Mutka, "Prophet Address Allocation for Large Scale MANETs", In Proceedings of the 22th IEEE Conference on Computer Communications (INFOCOM'03), March 2003.
- [6] Choi "Random and Linear Address Allocation for Mobile Ad Hoc Networks", IEEE Wireless Communications and Networking Conference(WCNC'05), 2005.
- [7] Li-Hsing Yen and Wei-Ting Tsai, "Flexible Address Configurations for Tree-Based ZigBee/IEE 802.15.4 Wireless Networks", 22nd International Conference on Advanced Information Networking and Applications(AINA), 2008